This is a sample session illustrating the use of certain Objectivity commands. It is not a tutorial per se, but you can repeat the steps here to see things in action for yourself. All the Objectivity commands accept the '-help' argument, which is normally enough to understand enough to use them.

In this session, I create a shallow copy of a federation, add and delete databases, and move them around in the filesystem. Finally, I have an example of removing stale locks.

The output of the Objectivity commands is somewhat abridged to save trees.

First I check that the AMS and lockserver are running on my desktop machine (pctony.cern.ch). The lockserver is there, but I have to start the AMS myself.

```
:~> oocheckams
AMS is not running on 'pctony'.
:~> oocheckls
The Lock Server is running on 'pctony'.
:~> oostartams
The AMS has been started (process ID = 29095).
```

Next I create a shallow copy of an existing federation on my desktop. I create a subdirectory for it, 'cd' into it, copy the bootfile and the federation file and use 'ooinstallfd' to make it into a real federation instead of just a pair of copied files.

The first attempt fails because I did not create the journal directory. The 'ooinstallfd' command will not make the directory for me. After creating the journal directory, 'ooinstallfd' succeeds.

```
:~> mkdir myfed
:~> cd myfed
:myfed> rfcp cmsuf01:/cms/reconstruction/user/jetDIGI0300/jetDIGI0300.boot .
272 bytes in 0 seconds through eth0 (in) and local (out)
:myfed> rfcp cmsuf01:/cms/reconstruction/user/jetDIGI0300/jetDIGI0300.FDDB .
9895936 bytes in 12 seconds through eth0 (in) and local (out) (805 KB/sec)
:myfed> ooinstallfd -fdnumber 2907 -lockserverhost pctony –jnldirpath \
/users/wildish/myfed/journal -nocatalog -nocheck \
/users/wildish/myfed/jetDIGI0300.boot
Updating Name Service values...
** System Error #3036: Lock Server cannot auto-recover FD.  Please verify that
                       the account running the Lock Server has permission to
                       access all the database files and the journal directory.
                       You may need to recover this FD manually.  Please check
                       the Lock Server log file.
** System Error #2502: Object Manager is unable to start a new transaction
** Error #2921: ooinstallfd : Unable to open the Federated Database
                "/users/wildish/myfed/jetDIGI0300.boot". Processing terminated.
** Error #2928: ooinstallfd : An error has occurred. Processing terminated.

:myfed> mkdir journal
```

```
:myfed> ooinstallfd -fdnumber 2907 -lockserverhost pctony -jnldirpath
/users/wildish/myfed/journal -nocatalog -nocheck
/users/wildish/myfed/jetDIGI0300.boot
Updating Name Service values...
Now updating System Name Space (catalog) values...
WARNING: Updating of Database File locations will be skipped...

Federated Database Installation complete.
```

The federation has been installed locally, as shown by 'oodumpcatalog'. I now have my shallow copy. If I add or create databases in this federation they will not be known to other federations.

```
:myfed> oodumpcatalog jetDIGI0300.boot | head -20
FD Name   = jetDIGI0300
FD ID     = 2907
FD File   = pctony::/users/wildish/myfed/jetDIGI0300.FDDB
Boot File = pctony::/users/wildish/myfed/jetDIGI0300.boot
Jnl Dir   = pctony::/users/wildish/myfed/journal
Lock Host = pctony

DB Name   = CARF_System.META.jetmetSignal0300
DB ID     = 11
DB Image  =
cmsuf01::/cms/reconstruction/user/jetDIGI0300/CARF_System.META.jetmetSignal0300.j
etmetHitPU.DB

DB Name   = LogBook.META.jetmetSignal0300
DB ID     = 12
DB Image  =
cmsb19::/shift/cmsb19/data0/reconstruction/userfeds/jetDIGI0300/./LogBook.META.je
tmetSignal0300.jetmetHitPU.DB
```

If I create a new database with the 'oonewdb' command, we can see that the file is created and attached to the federation. Again I use 'oodumpcatalog' to show this. Normally you would not create databases like this, CARF and ORCA between them will create the databases as needed.

After creating the database I use the 'oofile' command to check its attributes. I can get the pagesize and the DBID.

```
:myfed> oonewdb -db MyTest jetDIGI0300.boot
Created Database MyTest [DBID = 1903].
:myfed> oodumpcatalog jetDIGI0300.boot | grep -C 1903

DB Name   = MyTest
DB ID     = 1903
DB Image  = pctony::/users/wildish/myfed/MyTest.jetDIGI0300.DB

:myfed> oofile MyTest.jetDIGI0300.DB
Attempting to interpret the file...

Data format : Linux
```

```
Database format compatible with Objectivity release 4.0.10 through 5.x

File Type : Database
DB ID     : 1903
Page Size : 32768

You may only attach this Database file to a Federated Database with the
same Page Size and Schema.

**************************WARNING**************************
* You should verify that this file does not belong to an   *
* active Federated Database before moving or modifying it. *
***********************************************************
```

Now I delete the database from my federation. I use the 'oodeletedb' command, with the '-catalogonly' flag to ensure that the file is *not* deleted from the disk.

```
:myfed> oodeletedb -catalogonly -db MyTest jetDIGI0300.boot
Are you sure you want to delete the Database?
[Y-N]=> y

Deleted the Database "MyTest" (ID = 1903) from the Federated Database catalog.
```

'ls –l' shows the database file is still there, but the database entry has gone from the catalogue.

```
:myfed> ls -l
total 10158
-rw-rw-rw-   1 wildish  zh        1867776 Oct  8 17:45 MyTest.jetDIGI0300.DB
-rwxr-xr-x   1 wildish  zh        9895936 Oct  8 17:46 jetDIGI0300.FDDB
-rw-r--r--   1 wildish  zh            202 Oct  8 17:40 jetDIGI0300.boot
drwxr-xr-x   2 wildish  zh           1024 Oct  8 17:46 journal
```

Now I can re-attach the database to my federation. If I specified a different federation bootfile, I could attach my federation there instead (providing that DBID 1903 was not already taken!).

```
:myfed> ooattachdb -id 1903 -db MyTest -filepath
/users/wildish/myfed/MyTest.jetDIGI0300.DB -host pctony jetDIGI0300.boot
First pass through ...
  Attaching database...MyTest

Second pass through ...
  Attaching database...MyTest
```

Now for one of the ways to move the database file to a different directory. I create the directory, move the database file using standard unix commands, but then I have to tell the federation where to find it.

Using the 'oochangedb' command I specify the new filepath for this database in the federation catalogue. Again I have to use the '-catalogonly' flag to do this, because Objectivity runs a variety of checks and will complain if the file is not there.

Although you can use the '-catalogonly' flag (and others) with many commands to circumvent some of the checks that Objectivity performs, it is not a good idea to do so without a bit of reflection. There is a reason Objectivity checks things!

```
:myfed> mkdir mytags
:myfed> mv MyTest.jetDIGI0300.DB mytags/
:myfed> oochangedb -db MyTest –filepath \
/users/wildish/myfed/mytags/MyTest.jetDIGI0300.DB -host pctony jetDIGI0300.boot \
-catalogonly
:myfed> oodumpcatalog jetDIGI0300.boot | grep -C 1903
DB Name   = MyTest
DB ID     = 1903
DB Image  = pctony::/users/wildish/myfed/mytags/MyTest.jetDIGI0300.DB
```

Now for detecting and removing stale locks. Stale locks are locks that are left behind by processes that die abnormally. The easiest way to show this is to use 'ootoolmgr' to view the federation, and select 'Browse' from the 'Tools' menu. This will create two read locks. Then kill the 'ootoolmgr' process with the –KILL signal, and attempt to clean up the locks afterwards. Then use 'oolockmon' to see the locks. After I kill 'ootoolmgr' I use 'oocleanup' and give it the transaction ID to tell it which transaction to attempt to recover.

```
:myfed> ootoolmgr jetDIGI0300.boot &
[1] 29842
:myfed> oolockmon jetDIGI0300.boot
Lock Table for Lock Server on node "pctony" - your UID is 2907

    UID     HostID   PID    TransID  APID  Mode  Type FdbID  DbID OclID
  ------------------------------------------------------------------------
   2907     pctony 29842    1310719 65535  read   Ocl  2907     1     4
   2907     pctony 29842    1310719 65535  read   Ocl  2907     1     3

Lock table displayed, 2 entries.
```

The toolmgr PID is given as 29842. I kill it, brutally, and the locks remain…

```
:myfed> kill -KILL 29842
:myfed> oolockmon jetDIGI0300.boot
Lock Table for Lock Server on node "pctony" - your UID is 2907

    UID     HostID   PID    TransID  APID  Mode  Type FdbID  DbID OclID
  ------------------------------------------------------------------------
   2907     pctony 29842    1310719 65535  read   Ocl  2907     1     4
   2907     pctony 29842    1310719 65535  read   Ocl  2907     1     3

Lock table displayed, 2 entries.
```

The locks are still there, so I clean up with the transaction ID. ‘oolockmon’ then shows that the locks have been removed.

```
:myfed> oocleanup -transaction 1310719 jetDIGI0300.boot
Recovered transaction 1310719 for FD "jetDIGI0300.boot".

Finished processing all autonomous partitions.

Finished attempting to recover all specified transactions.

:myfed> oolockmon jetDIGI0300.boot
Lock Table for Lock Server on node "pctony" - your UID is 2907

    UID     HostID   PID    TransID  APID  Mode  Type FdbID  DbID OclID
  ------------------------------------------------------------------------

Lock table displayed, no entries.
```

I can use the ‘oocleanup’ command in another way. Instead of having to find the transaction ID for all the transactions that are in process, I can tell it to clean up all transactions that were started locally, i.e. on this host.

```
:myfed> ootoolmgr jetDIGI0300.boot &
[1] 29854
:myfed> oolockmon jetDIGI0300.boot
Lock Table for Lock Server on node "pctony" - your UID is 2907

    UID     HostID   PID    TransID  APID  Mode  Type FdbID  DbID OclID
  ------------------------------------------------------------------------
   2907     pctony 29854    1507327 65535  read   Ocl  2907     1     4
   2907     pctony 29854    1507327 65535  read   Ocl  2907     1     3

Lock table displayed, 2 entries.
:myfed> oocleanup -local jetDIGI0300.boot
Recovering local transactions for FD "jetDIGI0300.boot" on host "pctony".
** Error #3104: oocleanup: Failed to recover transaction 1507327 for FD
               "jetDIGI0300.boot". Error #0: .

Finished processing all autonomous partitions.

Finished attempting to recover all specified transactions.

:myfed> oolockmon jetDIGI0300.boot
Lock Table for Lock Server on node "pctony" - your UID is 2907

    UID     HostID   PID    TransID  APID  Mode  Type FdbID  DbID OclID
  ------------------------------------------------------------------------

Lock table displayed, no entries.
```

That’s all for now. Don’t forget the ‘-help’ argument, and, of course, those fine manuals.